

Synthèse en Français

PWA

Les Progressive Web Applications
(PWA) vont-elles remplacer les
applications traditionnelles ?

Julien CUÉNOT

I - Présentation générale

1.1 - Introduction

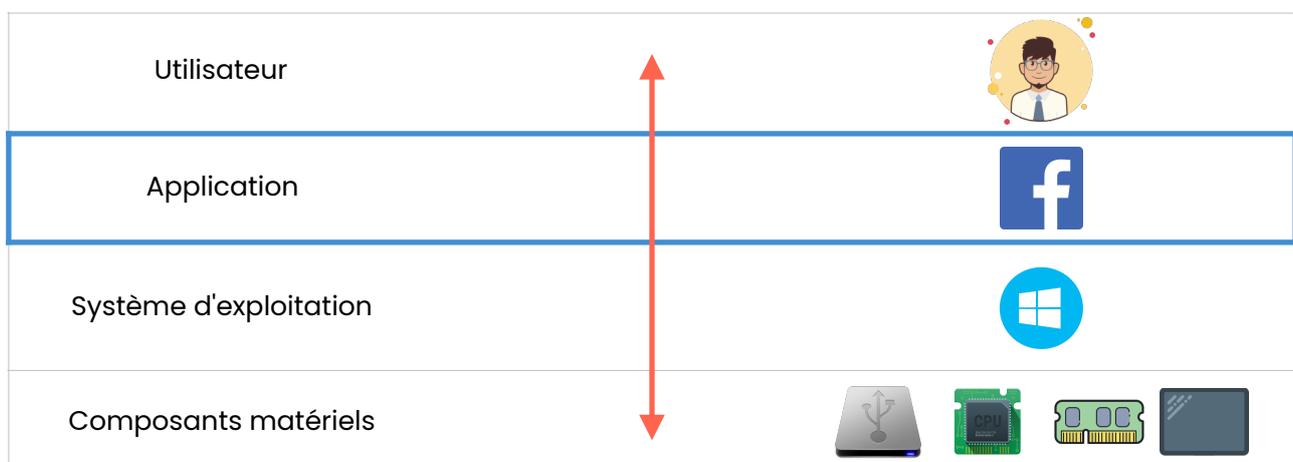
Nous utilisons en permanence des applications. Elles peuvent être aussi bien installées sur nos smartphones, tablettes, ordinateurs fixes et mobiles, montres, ou encore sur des robots explorateurs allant sur mars. Bien que nous n'ayons pas tous un robot Martien, les applications représentent ce qu'est l'informatique au sens large. Sans application(s) rien ne se passerait dans l'électronique de nos appareils.

Aujourd'hui se repérer au milieu de tous les dispositifs et des applications qui se présentent sous différentes formes est un véritable casse-tête.

1.2 - Qu'est-ce qu'une application ?

L'application est la couche informatique la plus haute que l'utilisateur va utiliser. Grâce à l'application, l'utilisateur va (sans s'en apercevoir) utiliser les différents composants de son appareil et les mettre à profit pour une tâche qu'il souhaite réaliser.

Rappelons que les 3 termes "Application", "Logiciel" et "Programme" désignent la même chose, bien que par convention nous les utilisons dans des contextes différents. Une Application est souvent utilisé comme diminutif de "application pour mobile" mais rien ne diffère avec un "logiciel" et un "programme". Le terme logiciel est souvent associé à une solution plus lourde destinée aux ordinateurs fixes et portable, les programmes, eux, plutôt dans l'automatisme industriel et l'IOT. Mais les 3 termes sont interchangeables et dans la suite du document seul le terme "application" sera utilisé.



Note : Le système d'exploitation est une application, mais ne complexifions pas les choses :-)

II - Les différents types d'applications

2.1 - Les 2 grandes familles d'applications

Il existe 2 familles d'applications différentes qui sont les applications '**natives**' et les applications '**web**'. Ces applications sont différentes tant dans la façon dont elles sont développées, que dans la manière dont elles fonctionnent.

Différents points peuvent motiver le choix d'un type d'application :

- Les fonctionnalités disponibles (Bluetooth, disponibilité hors ligne)
- Les performances (Rapidité, fluidité)
- L'adaptabilité aux différents appareils (Taille d'écran, OS, processeurs)
- Coûts (Développement initial et maintenance)
- Canaux de distribution (Par le web, par un store d'applications certifiées, ...)



Applications de type "natif"		Applications de type "web"	
Applications Natives	Applications Natives hybrides	Applications Web Progressives	Applications Web
Description : C'est une application classique, téléchargée puis installée sur l'appareil.	Description : Similaire à l'application native, sauf qu'elle emprunte des technologies issues des applications web et facilite l'export vers différents dispositifs.	Description : C'est une application web qui emprunte des fonctionnalités issues des applications natives.	Description : Un site web dédié à une application métier.
Exemple : L'application Twitter de l'app store.	Exemple : ...	Exemple : La web app de Twitter, une fois installée en local, est une PWA.	Exemple : La web app Twitter.



2.2 - Les applications natives

Une **application native** est développée spécifiquement pour un des systèmes d'exploitation visé (MacOS, Ubuntu, Windows, iOS, Android, etc). Les **applications natives** pour iPhone sont par exemple développées avec le langage de développement Swift et utilisent généralement des bibliothèques disponible uniquement sur iOS.

Inconvénients :

✗ Appareils de formats différents

Ce qui complexifie l'application ou entraîne le développement d'applications différentes.

✗ Systèmes d'exploitation différents

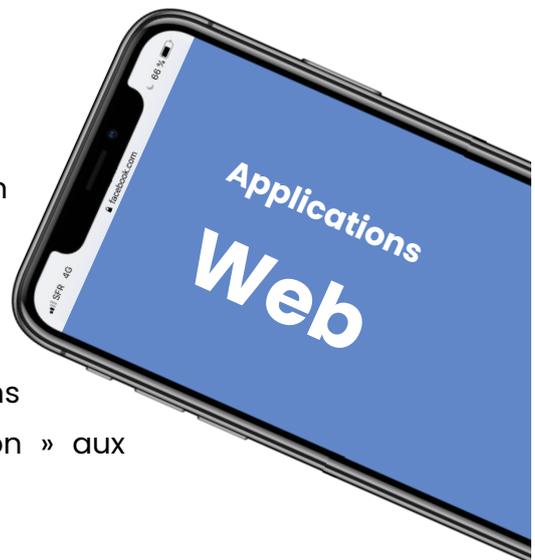
Une application pour Windows 10 ne fonctionne pas avec un iPad sous iOS 13.

✗ Mises à jour des systèmes d'exploitation

Elle entraîne souvent des bugs, ou empêche totalement l'application de fonctionner.

2.3 - Les applications web

Une **application web** est une application développée à la manière d'un site web, accessible et exécutable par le biais d'un navigateur Internet. Un webmail, un service bancaire en ligne ou un moteur de recherche sont donc des applications web. Les "**applications web**" sont en « opposition » aux "applications natives".



Inconvénients :

✗ Pas de hors ligne

L'application web n'est donc pas utilisable en l'absence de connexion internet.

✗ Pas de NFC

Les fonctionnalités NFC de l'appareil utilisé, souvent un mobile, ne sont pas accessible par l'intermédiaire d'une application web classique.

✗ Pas de Bluetooth

Il est impossible, pour une application web classique, de se connecter à des périphériques bluetooth.

III - Les Progressive Web Applications

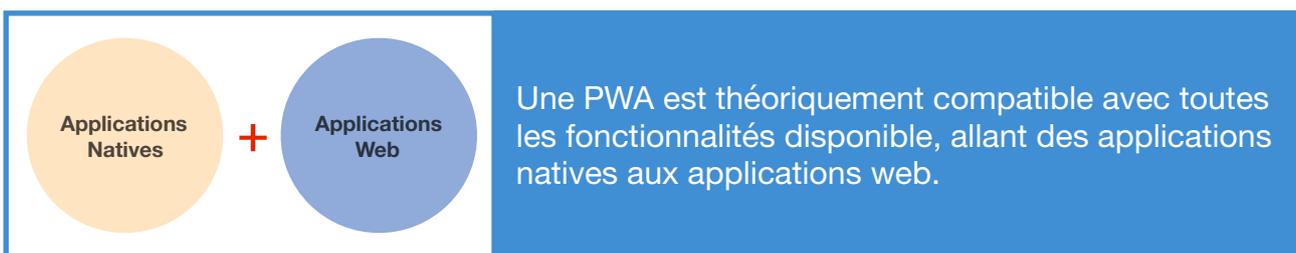
Introduction

Une application web progressive (PWA ou Progressive Web App) est une application se situant techniquement à « mi-chemin » entre un site ou application web et une application native « traditionnelle ».



Fonctionnalités

Une progressive web app (PWA) peut être définie par simplification comme une forme d'application web (ou de site web) qui peut être accessible directement à partir d'un navigateur internet et qui peut, après autorisation et installation, être utilisée comme une application native avec les fonctionnalités avancées du dispositif (appareil photo, géolocalisation, Bluetooth, etc).



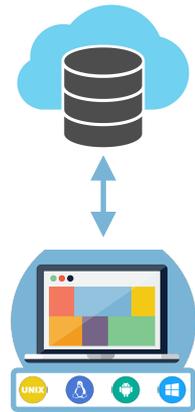
Système d'exploitation

La PWA est avant tout une application web. Elle est donc techniquement compatible avec n'importe quel système d'exploitation disposant d'un navigateur internet. Lorsque la PWA est installée sur l'appareil, elle s'exécute par le navigateur web mais de manière invisible pour l'utilisateur. C'est à dire sans la barre de navigation ou autres éléments d'interface propre au navigateur.



Une seule et unique application ★

La PWA est avant tout une application web. C'est à dire qu'elle utilise les mêmes composantes logicielles et matérielles. Comme représentée par le schéma ci-contre, la PWA fonctionne au sein du navigateur internet (Safari, Firefox, Chrome, Edge...) qui viens exécuter l'application.



- Internet
- Serveur web distribuant une application
- données
- Navigateur internet
- Système d'exploitation

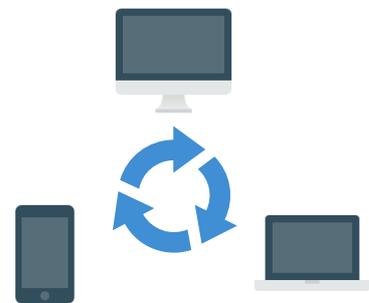


Web mais hors-ligne tout de même ★

C'est l'innovation numéro 1. Une fois installée, la PWA peut être d'exécutée sans nécessiter une connexion internet. Et ce, tout en étant une application de type 'web'.

Synchronisation simplifiée ★

La synchronisation est au coeur même de la Progressive Web App. C'est à dire que si vous devez bâtir une application qui doit pouvoir fonctionner à la fois en ligne et hors ligne, que cette même application doit synchroniser ses données en ligne après avoir été utilisée hors ligne, alors la PWA est très certainement la solution la plus adaptée. En effet cette méthode de fonctionnement le fondement même des PWA.



Le meilleur des deux mondes ★

La PWA en étant une application de type web dispose de divers avantages tel que d'être compatible avec n'importe quel appareil. Mais elle n'a rien à envier aux applications natives car elle dispose de fonctionnalités qui leurs sont habituellement réservées.

IV – Étude de l'existant

4.1 – La problématique d'aujourd'hui

La problématique à laquelle nous sommes confrontés aujourd'hui dans le domaine des applications, est qu'il faut prendre une décision sur quel type d'application choisir. Pourtant, les 2 types d'applications "**web**" et "**natives**" répondent pour chacun à des besoins différents, et possèdent chacun leurs avantages et inconvénients. Il convient donc d'avoir une vision globale de ce qui se fait aujourd'hui pour prendre de bonnes décisions techniques, lors d'un projet ou d'une stratégie d'entreprise.

4.2 – Vers une entité unique ?

"La PWA semble être la solution à toutes les emmerdes"

Source : Un point fait par l'EBG Paris sur le potentiel de la PWA à l'automne 2017 : <https://youtu.be/F0Np9yd3x-I>

La situation actuelle vis-à-vis des applications

Bien que très excitant et intéressant, le monde des applications est un casse-tête de plus en plus compliqué à appréhender. Il existe différents types d'application, possédant chacun ses spécificités. Après avoir eu l'idée et avoir pensé à comment la transformer en application réussite, il faut avoir en tête que chaque type d'application ne s'adresse pas forcément aux mêmes besoins. C'est pourquoi il est important de bien cerner les différents types d'applications.

Quelle décision technique ?

Afin d'effectuer un choix technique pertinent, il est indispensable d'avoir une vision ainsi qu'une compréhension globale de ce qu'est une application native, hybride, web, progressive, mobile, desktop, responsive.... Le casse-tête commence.

Vers une homogénéisation des applications

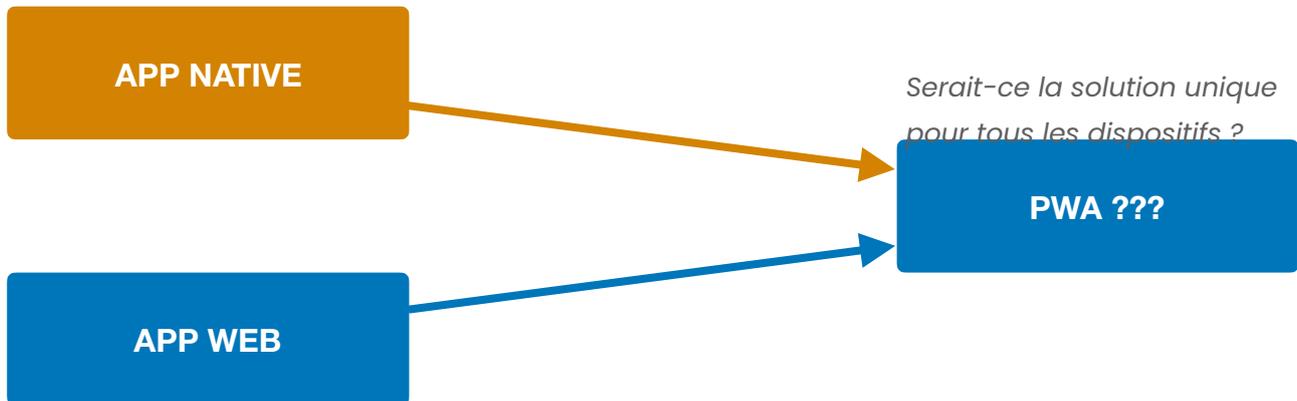
Aujourd'hui, le choix technique de la solution ne doit pas être pris à la légère. Les applications de type 'web' se rapprochent des applications de type 'native' et inversement. Les fonctionnalités et les performances sont maintenant similaires.

Risque à court terme limité

On peut donc se dire que le risque à court terme est limité, étant donné que l'on pourra aller là où l'on veut avec n'importe quel type d'application. C'est vrai. Mais à plus long terme la donne change.

Risque à long terme

Là où le risque est bien présent, c'est sur la pérennité à long terme et si le ROI (retour sur investissement) est prévu pour être long. En effet, les différents types d'application semblent se diriger vers une entité unique, qui semble être la PWA.

**La PWA est une application web ++**

C'est là qu'est toute la question. Y aura il une entité commune ? Est-ce que l'entité commune sera une PWA ? Les indicateurs disent oui.

Suivant ce raisonnement, pour une solution à long terme, les applications web, ou mieux les PWA, semblent être des solutions idéales car leur évolution dans le temps sera assurée par le biais de mises à jour, sans avoir à tout redévelopper depuis zéro.

V - Étude de cas

5.1 - Présentation du projet TRACY

En 2015, DAVI a développé une solution nommée "TEMIS" pour le client KREMER. Aujourd'hui renommée "TRACY", cette solution a pour objectif de faciliter le processus de contrôle régulier des équipements industriels. Elle permet de tracer et de suivre le cycle de vie d'un équipement de manière fiable et moins contraignante, par l'utilisation de tags NFC placés sur les équipements et qui permettent leur identification.

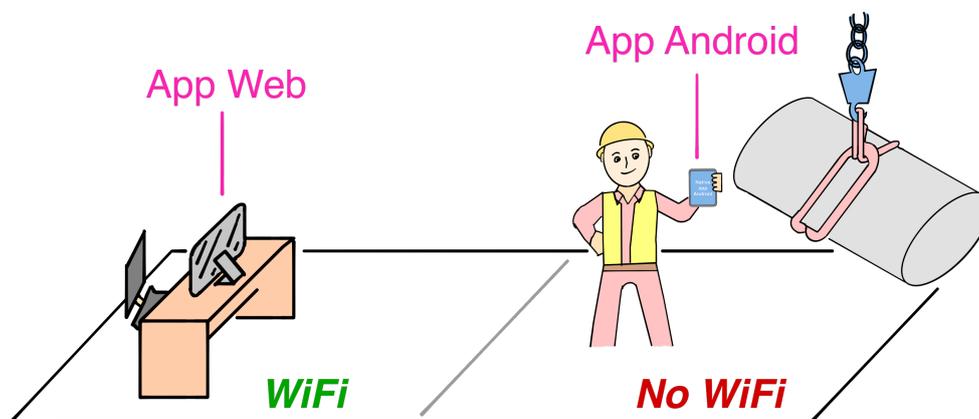
5.2 - Expression du besoin

Aujourd'hui, l'architecture rigide mise en place pour la solution TRACY doit être re-développée. C'est l'occasion de se renseigner sur les solutions alternatives et voir si il est possible de recourir aux PWA.

5.3 - Existant : Les applications

La solution de contrôle, pour répondre aux besoins des clients doit être utilisable même en l'absence de connexion internet. En effet, plusieurs équipements dans les entreprises des utilisateurs de l'application, sont situés à des emplacements où une connexion internet n'est pas possible ou trop contraignante à mettre en place. Pour répondre à ce besoin, le choix technique a été de passer par l'utilisation de 2 applications.

- La première est une application web basée sur le CMS Drupal. Cette application nécessite une connexion internet pour être utilisée. Elle permet d'administrer l'ensemble des données.
- La seconde application est une application native Android qui elle peut être utilisée même en l'absence de connexion internet. Elle permet aux contrôleurs d'aller contrôler les équipements (d'une usine par exemple) même lorsque la connexion internet n'est pas disponible.



4.4 - Une PWA peut elle répondre aux problématiques de TRACY ?

Si nous reprenons la solution TRACY dans son ensemble, deux applications différentes ne sont pas un besoin mais une nécessité liée aux verrous technologiques actuels. En effet, si l'on reprend le tableau ci-dessous, une application web seule ne peut pas répondre à tous les besoins. Une application native seule ne le peut pas non plus. C'est la raison pour laquelle aujourd'hui deux applications doivent être maintenues et complexifiées le Système d'Information.

La question qui s'est posée était de savoir si une PWA pouvait être l'unique et meilleure solution pour répondre à tous les besoins de cette solution, d'un point de vue global, dans son ensemble.

**** Le tableau ci-dessous illustre la réponse de manière théorique ****

Problématiques fonctionnelles globales de la solution	La PWA est-elle la solution la plus adaptée ?
L'expérience utilisateur doit être intuitive, rapide et fluide.	
La charte graphique doit être moderne (s'appuyer sur l'interface actuelle de l'application Android).	
L'application doit pouvoir se synchroniser avec divers appareils et fournir des données cohérentes à l'ensemble de la flotte d'appareils.	
La solution, tout ou partie, doit pouvoir fonctionner avec ET sans connexion internet.	
L'application doit pouvoir se connecter à des périphériques Bluetooth.	
L'application doit pouvoir envoyer des notifications.	
L'application est une solution "multi-tenants". C'est à dire qu'elle doit gérer intégralement les aspects "Revendeurs de la solutions", "Clients de la solution" et "Utilisateurs finaux", avec différents niveaux de droits.	
ETC...	

4.6 - La solution PWA a t-elle été retenue pour la prochaine version de TRACY ?

✗ Et non

Non, la solution PWA n'a pas été retenue (dans l'immédiat) pour ce projet. Bien que la PWA coche théoriquement toutes les cases, la jeunesse de cette technologie est un frein indéniable. Dans un contexte de mise en production début 2020 et où le droit à l'erreur n'est pas permis.

✗ Tout juste pleinement supportées

À l'heure où j'écris ces lignes, seul le navigateur Google Chrome pour Android et ordinateurs de bureau supportent toutes les fonctionnalités proposées par les PWA, comme le fonctionnement hors-ligne. Du côté d'Apple, la prochaine mise à jour du système d'exploitation mobile apporte son lot d'améliorations, avec une prise en charge totale des PWA. iOS 13 dont je possède un accès en avant première car je suis développeur Apple certifié, ne sort pour le grand public que dans quelques jours pour le grand public !

✗ Manque de recul

Les PWA sont tellement récentes en méconnues que très peu de développeurs web ne connaissent leur intérêt. Faites une expérience sociale et demandez à votre développeur si il les connaît et en quoi elles consistent. Vous serez surpris des réponses. Si c'est un développeur web vous égaillerez probablement sa journée en lui apprenant qu'elles existent 😊. Cependant, ce manque de recul engendre certains freins comme un manque de documentation disponible sur internet et oblige aujourd'hui les développeurs à se reposer uniquement sur les documentations officielles (ex: Google), sans pouvoir compter sur l'aide et les réponses rapides provenant des forums de discussion (ex: Stack Overflow). Aussi, le manque de retours rend plus difficile la mise en place d'un plan de gestion des risques.

✗ Difficulté à trouver des ressources compétentes

Ce qui peut sembler être une solution d'avenir pour l'entreprise prestataire et représenter un coût en maintenance beaucoup plus faible peut aussi rapidement devenir contre-productif si l'architecture logicielle n'est pas correctement conçue.

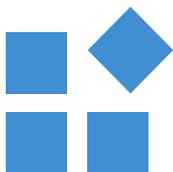
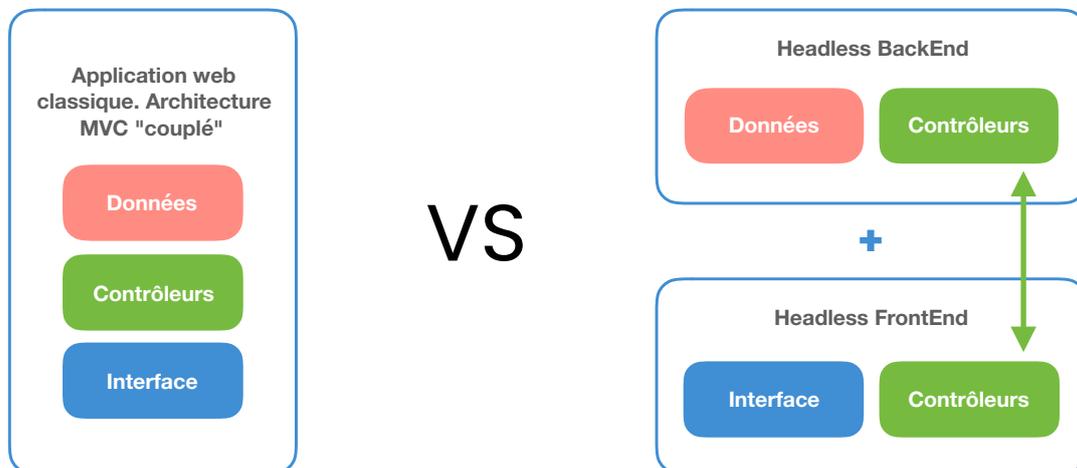
Tarif ✗

La part d'inconnu liée aux PWA fait perdre du temps et impact directement le coût de la solution. De plus il est fortement recommandé de passer par un architecte web qui dessinera l'architecture dite "offline-first" (qui permet un fonctionnement hors-ligne pertinent) avant que les développeurs ne puissent commencer à créer des briques logicielles pour la solution.

4.7 - Une architecture "Headless", prête à délivrer une PWA !

Introduction

La solution technique choisie pour la nouvelle version de TRACY est une architecture dite "Headless". C'est une architecture logicielle qui repose sur la philosophie MVC (Model View Controller) mais en poussant le raisonnement à l'extrême. À la différence d'une architecture MVC classique, l'architecture Headless sépare les parties gestion des données et l'interface utilisateur à tel point que ce sont 2 applications différentes qui communiquent à l'aide d'une API. L'API est une brique logicielle permettant d'inter-connecter des applications entre-elles.



Modularité et maintenance facilitée

L'architecture Headless permet une grande modularité des briques logicielles pour les évolutions futures et permet de faciliter la maintenance. C'est une solution dite "scalable".

Faciliter une future transition vers une PWA

Se lancer dans le développement d'une PWA aujourd'hui peut être cher et contraignant sur plusieurs aspects, mais ces inconvénients devraient disparaître d'ici quelques années. Le choix de l'architecture Headless est en partie lié au fait d'une possible future transition vers une PWA, lorsque les fonctionnalités, les retours et les documentations seront parfaitement au point.

V - Conclusion

5.4 - Un avenir très prometteur à court terme pour les

PWA

Aujourd'hui, je manque de recul sur l'utilisation des PWA comme unique application en production ce qui rend difficile de recommander ce type de solution. Les PWA sont toujours en cours de développement et l'ensemble des environnements n'est toujours pas homogène. Comme nous avons pu le voir, à l'heure où j'écris ces lignes, Apple ne prendra pleinement en charge les PWA que dans quelques jours, lorsque la mise à jour dont je dispose, iOS 13 (en beta développeur), sortira pour le grand public.

Néanmoins, les PWA semblent répondre à une demande générale d'homogénéisation des applications, des fonctionnalités et des méthodes de développement.

La solution que je recommande aujourd'hui dans mon entourage est une architecture web "Headless" qui permet une modularité et scalabilité incomparable aux autres types d'architecture. En plus des nombreux avantages précédemment expliqués, l'architecture Headless permet de très facilement passer d'une application web classique à une PWA, ce qui sera probablement le cas d'ici quelque années pour le projet TRACY.